УДК 004.89

Т. В. Ермоленко
Государственное образовательное учреждение высшего профессионального образования
«Донецкий национальный университет»
83000, г. Донецк, пр. Театральный, 13

КЛАССИФИКАЦИЯ ОШИБОК В ТЕКСТЕ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ

T. V. Yermolenko
State Educational Institution of Higher Education «Donetsk National University»
83000, c. Donetsk, Teatralnyiy av., 13

CLASSIFICATION OF ERRORS IN THE TEXT BASED ON DEEP LEARNING

Т. В. Єрмоленко
Державна освітня установа вищої професійної освіти
«Донецький національний університет», м. Донецьк
83000, м. Донецьк, пр. Театральний, 13

КЛАСИФІКАЦІЯ ПОМИЛОК У ТЕКСТІ НА ОСНОВІ ГЛИБИННОГО НАВЧАННЯ

В статье рассматривается задача классификации ошибок, сделанных в русскоязычных текстах, с помощью глубоких нейросетей. Для решения данной задачи была выбрана архитектура QRNN с применением слоя внимания, вместо векторных представлений слов был использован предпоследний слой из обученных языковых моделей. Языковые модели были обучены с помощью рекуррентной сети с архитектурой LSTM. Численные исследования показали эффективность предложенной архитектуры, точность классификации ошибок на тестовой выборке превысила 95%.
Ключевые слова: автоматическая обработка естественного языка, рекуррентные сети, квазирекуррентные нейронные сети.

The article considers the task of classifying errors made in Russian-language texts using deep neural networks. To solve this problem, the QRNN architecture was selected using the attention layer; instead of vector representations of words, the penultimate layer from the trained language models was used. Language models were trained using a recursive network with LSTM architecture. Numerical studies have shown the effectiveness of the proposed architecture, the accuracy of the classification of errors in the test sample exceeded 95%.
Key words: natural language processing, recurrent neural networks, quasi-recurrent neural networks.

У статті розглядається задача класифікації помилок, зроблених в російськомовних текстах, за допомогою глибинних нейромереж. Для вирішення даної задачі була обрана архітектура QRNN із застосуванням шару уваги, замість векторних представлень слів був використаний передостанній шар з навчених язкових моделей. Язикові моделі були навчені за допомогою рекурентної мережі з архітектурою LSTM. Чисельні дослідження показали ефективність запропонованої архітектури, точність класифікації помилок на тестовій вибірці перевищила 95%.
Ключові слова: автоматична обробка природної мови, рекурентні мережі, квазірекурентні нейронні мережі.

Автоматическое исправление ошибок в естественно-языковых текстах – сложная и до сих пор не решенная задача, относящаяся к области автоматической обработки естественного языка (NLP – Natural Language Processing).

Текст, подвергшийся корректуре, удобен для его дальнейшей автоматической обработки (например, в системах по извлечению фактов, при обработке поисковых запросов), наличие встроенной системы автоматической коррекции орфографии в текстовых редакторах и мобильных устройствах несомненно облегчают набор текста для пользователей. NLP-Системы на разных уровнях (морфологическом, синтаксическом, семантическом) включают в себя модуль коррекции ошибок.

Автоматическое детектирование и исправление ошибок – важная часть систем машинного перевода, определения авторства, программ для изучения языков, приложений с голосовым вводом и выводом.

Современные модели, обученные на значительном объеме данных, отбирают кандидатов на исправление, основываясь на вероятностных методах, а не на правилах, что делает их более универсальными. Тем не менее, для русского языка качество работы таких систем далеко от идеала. Особенно, если принимать во внимание растущее число текстов в социальных медиа (блогах, социальных сетях, форумах), далеких от нормативного языка.

Системы автоматического исправления ошибок, помимо больших словарей, используют n-граммные языковые модели, построенные на больших текстовых корпусах, дополненные моделями, учитывающими в том или ином виде ошибки (учитывались вероятности различных перестановок, замен, вставок и удалений, иногда в зависимости от контекста), и позиционными правилами (фонетические ограничения на начало слова, конец слова, сочетания согласных и гласных, характерные для каждого конкретного языка). Для отбора кандидатов на исправление наряду с расстоянием Левенштейна (обычно не более 3) используются скрытые марковские модели. Это позволило исправлять простые опечатки, но проблемой остались словарные опечатки и ошибки, неисправимые без учета контекста [1].

На сегодняшний день для решения этой задачи используют новейшие вычислительные методы, что в первую очередь связано с ростом вычислительных мощностей и доступностью больших корпусов текстовых данных. Бурное развитие методов машинного обучения, нейронных сетей позволило использовать этот мощный аппарат для NLP-задач, в частности, для задач поиска, классификации и исправления ошибок.

Системы автоматического исправления ошибок на основе нейросетей выполняют обработку текста последовательно за несколько этапов:

1. Токенизация (разбиение на орфографические слова) входного текста.
2. Кодирование полученных токенов.
3. Детектирование и классификация ошибок.
4. Исправление ошибок.

Для получения качественной языковой модели и использования глубокого обучения необходимы большие объемы данных. Как правило, имеющихся данных для обучения системы исправления текстовых ошибок недостаточно, поэтому используют стратегию аугментации – создания дополнительных обучающих данных из имеющихся.

В данной работе основное внимание уделено описанию модели детектирования и классификации ошибок в тексте на основе глубокого обучения.

Классификация ошибок, допускаемых в тексте

Классы ошибок в тексте можно разделить на 4 категории:

1) морфологические: нехарактерная для языка наблюдаемая последовательность морфологических категорий, неправильное образование форм слов различных частей речи (например, «докторы» вместо «доктора»; «яблоков» вместо «яблок»);

2) орфографические, вызванные неправильным написанием слов (например, «извини» через «е»);

3) синтаксически-пунктуационные: неправильная постановка знаков препинания, неверное соединение слов в словосочетаниях и предложениях, пропуск тире, ошибки в согласовании и управлении;

4) лексические: употребление похожих по звучанию слов (например, «одинарный» вместо «ординарный»), путаница в близких по значению словах (например, «абонент» вместо «абонемент»), непреднамеренное образование новых слов.

В табл. 1 – 4 приведена детальная классификация ошибок различных категорий, используемая при разработке системы. Лексические ошибки разработанной моделью не детектировались.

Таблица 1 – Классы морфологических ошибок

Обозначение класса	Описание ошибки	Пример
Misspell	комплексная ошибка, затрагивающая несколько букв (трансформировано написание слова в целом)	*деяк* (вместо *денег*)
Infl	использование окончания, которое отсутствует в парадигме данного слова	*я с Кодием* [‘=*Коди*’] *будем развиться*; *в статье не говорять о побочним действиях нелегальных наркотиках*
Num	употребление слова в неверной числовой форме (несоответствующей контексту или аномальной для этого слова)	*мы занимались физкультурами*
Gender	изменение родовой принадлежности слова	*автор даёт серьезныю комментарию об американском обществе*
Altern	ошибка в чередовании основы	*любю* (вместо *люблю*)

Таблица 2 – Классы орфографических ошибок

Обозначение класса	Описание ошибки	Пример
Graph	смешение алфавитов	*Друг* [‘=*друг*’], *геагировала* [‘=*реагировала*’]
Ortho	прочие орфографические ошибки	*карова*, *прожыть*
Translit	неверная транслитерация имени собственного	*Гиминьгвэй* [вместо *Хемингуэй*]

Таблица 3 – Классы синтаксически-пунктуационных ошибок

Обозначение класса	Описание ошибки	Пример
Tense	ошибка во временной форме глагола	-
Refl	неверное употребление возвратных глаголов	*стречала* [= *встречалась*]
AgrNum	ошибка в согласовании по числу	*друг слышал и нравится что вы сказал* [‘=*сказали*’]
AgrCase	ошибка в согласовании по падежу (не используется в случаях нарушения падежного управления)	*с этом значением*
AgrGender	ошибка в согласовании по роду	*Наша дружба самая важная аспект моей жизни*
AgrPers	неверное согласование сказуемого по лицу	*она никогда не буду стать* [‘=*станет*’] *врачом*
AgrGerund	ошибка в согласовании деепричастия	*Ходя по этому городу мой рот не закрывался от удивления*
Gov	ошибка в управлении	*полон красотой* [вместо *полон красоты*]; *найти хороших друзья* [вместо *найти хороших друзей*]
WO	ошибка в порядке слов	*с большими трудностями приходится сталкиваться директору приюта ежедневно*
Brev	Употребление кратких/полных прилагательных	*чай был такой вкусен*! [вместо *такой вкусный*]
Com	ошибки в сравнительных формах	-
Syntax	прочие синтаксические ошибки, не описанные в классификации	-
Constr	неверный выбор конструкции или ошибка внутри конструкции	*я имею две собаки* [вместо *у меня есть*]; *на большей части моей жизни я не знала, что такое друг*

Таблица 4 – Дополнительные классы ошибок

Обозначение класса	Описание ошибки	Пример
Extra	вставка ненужного элемента (буквы, морфемы или слова)	*У него ещё одна причина, чтобы увидеть, если ли она достойна услышать его теорию* [лишний союз *если*]
Transp	перестановка двух соседних элементов (буквы, морфемы или слова)	-
Subst	замена буквы, морфемы или слова	-

Построение языковой модели

Для обучения языковой модели и модели классификации ошибок были подготовлены следующие наборы данных:

1) словарь, состоящий из 7 млн уникальных словоформ;

2) текстовый корпус, на котором обучалась языковая модель, состоящий из текстовых массивов, собранных из новостных сайтов объёмом в 208,006,138 слов, общий размер около 15 Гб;

3) текстовый корпус, состоящий из дорожки SpellRuEval [2], а также данных, полученных из «Русского учебного корпуса» [3], на котором обучалась модель классификации. Выборка была поделена на обучающую и тестовую в отношении 95/5.

Как правило, при обучении нейросетевых моделей для NLP-задач используют векторные представления слов, наиболее распространенными из них являются:

- векторно-пространственные модели на основе матрицы встречаемости слов (модель мешка слов – BoW, tf–idf [4], латентно-семантический анализ);
- модели, учитывающие контекст (word2vec [5]; FastText [6]);
- модель GloVe, сочетающая сильные стороны двух вышеуказанных подходов [7].

Векторные представления слов обладают рядом недостатков, среди которых можно отметить:

- ограниченную словарную базу, что исключает обработку внесловарных слов;
- предположение о независимости каждого вектора, что приводит к плохой реализации семантических отношений между словами;
- отсутствие интерпретируемости компонент построенных векторов.

В связи с этим как альтернативу векторному представлению слов предлагается использовать предпоследний слой, так называемый fine-tune, из обученных языковых моделей.

В данной работе для обучения языковой модели была использована архитектура ULMfit [8], которая основана на модели weight-dropped LSTM, отличающаяся от LSTM наличием слоя DropConnect, применяющегося для регуляризации рекуррентных соединений [9].

Процесс обучения ULMfit состоит из трёх этапов. На первом этапе (рис. 1 a, LM pre-training) языковая модель обучается на общем наборе данных общим характеристикам языка и семантическим отношениям между словами.

На втором этапе обучения (рис. 1 b, точная настройка LM) выполняется точная настройка (fine-tune) для модели, полученной на первом этапе, с использованием дискриминационной тонкой настройки и наклонных треугольных скоростей обучения (slanted triangular learning rates, STLR), чтобы заставить модель изучать специфические особенности, определяемые набором входных данных.

На третьем этапе (рис. 1 c) выполняется точная настройка модели. Предварительно обученная модель дополняется двумя стандартными слоями прямого распространения и softmax-нормализацией на последнем слое. Выполняется STLR для сохранения представлений низкого уровня и адаптации к представлениям высокого уровня.

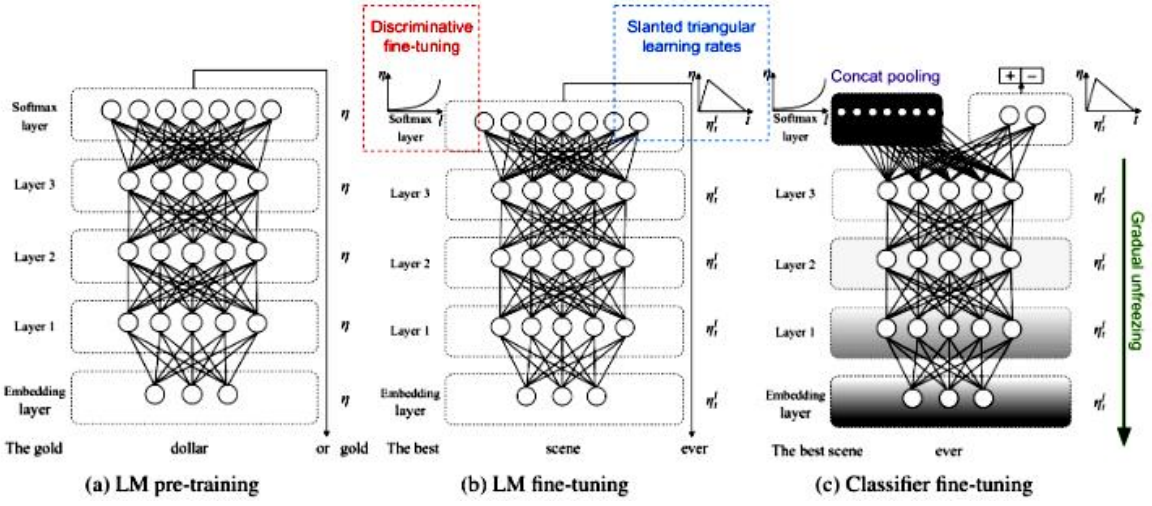

Рисунок 1 – Этапы обучения модели ULMfit

Архитектура ULMfit была выбрана в силу своей универсальности, поскольку она может работать с документами различных типов и наборами данных различной длины, не требует настройки пользовательских функций для обеспечения совместимости с другими задачами. Дополнительно ULMfit может быть улучшена с помощью добавления слоя с вниманием и дополнительных внутренних слоёв.

Для обучения модели ULMfit был использован алгоритм кодирования фраз, который, в первую очередь, направлен на оптимизацию процесса обучения модели при помощи мини-пакетного типа обучения [10].

Для кодирования слов использовался словарь, содержащий около 7 млн уникальных словоформ, знаки препинания, а также метки «BOS» (начало фразы), «EOS» (конец фразы), «PAD» (токен для разделения предложений), «UNK» (токен для обозначения внесловарного слова). В случае если слово w отсутствует или его частота появления в тексте меньше заданного порога ($thresh_w$) w=«UNK». Иначе из словаря извлекается id слова. В итоге получается массив индексов слов $U=\{u_m\}$.

Каждое слово, в свою очередь состоит из индексов символов, в него входящих. Таким образом, на вход сети поступает тензор индексов.

При трансформации данных для мини-пакетного типа обучения стоит помнить о разных длинах фраз в массивах данных и слов. Чтобы разместить фразы разных размеров в одном пакете, необходимо сделать матрицу E длины $L_{\max} \cdot b_s$ ($L_{\max}$ – максимальное количество слов в фразе, b_s – размер пакетов), где фразы короче $L_{\max}$ должны быть дополнены нулями после индекса «EOS». Однако, если просто преобразовать фразы в матрицы путем преобразования слов в их индексы и сделать нулевое заполнение, то тензор будет иметь форму (b_s, $L_{\max}$), и при индексировании первого измерения будет возвращаться полная последовательность по всем временным шагам. Необходимо иметь возможность индексировать пакет по времени и по всем последовательностям в пакете, поэтому, чтобы индексирование по первому измерению возвращало шаг по времени для всех предложений в пакете, выполняется транспонирование матрицы E. (рис. 2).

Рисунок 2 – Операция преобразования матрицы индексов слов для мини-пакетного обучения

При обучении модели ULMfit были использованы следующие параметры нейросети: количество слоёв 3, функция активации – ReLU, функция оптимизации – adam, количество эпох 5, размер пакетов – 64, функция потерь – кросс-энтропия (loss cse), размер скрытого слоя – 512, коэффициент dropout = 0.1.

В качестве векторного представления слов (word embedding) использовались выходы 100 нейронов предпоследнего слоя.

Построение модели классификации ошибок в тексте

RNN, LSTM являются мощными инструментами для моделирования данных в виде последовательности, но зависимость вычислений каждого временного шага от выходного сигнала предыдущего временного шага ограничивает параллелизм и делает их громоздкими для очень длинных последовательностей. Для решения этой проблемы используют квазирекуррентные нейронные сети (Quasi-Recurrent Neural Networks, QRNN). Этот подход к моделированию нейронных последовательностей чередует сверточные слои, применяющиеся параллельно через временные шаги, используя минималистическую функцию рекуррентного пула, которая используется параллельно по каналам [11]. Таким образом, преимуществом QRNN является тот факт, что сеть работает в 16 раз быстрее, чем рекуррентные; в то же время показывая такую же точность классификации (рис. 3).

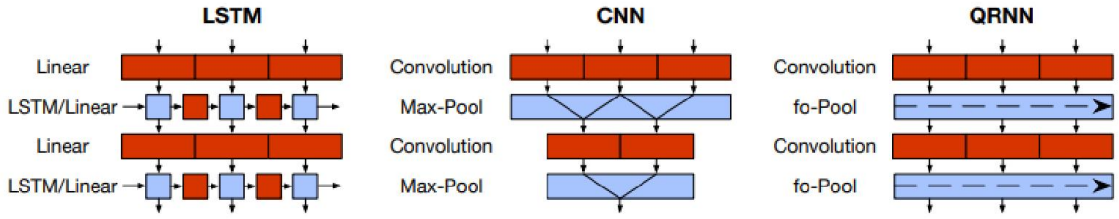

Рисунок 3 – Диаграммы блоков, показывающие вычислительные структуры LSTM, CNN, QRNN (красные блоки – операция перемножения матриц, непрерывный блок означает, что эти вычисления могут выполняться параллельно; синие – функции без параметров, которые работают параллельно вдоль измерения канала)

В данной работе для классификации ошибок в тексте была выбрана архитектура QRNN с применением слоя внимания, которая отображена на рис. 4, где l – длина строки; d – количество строк; N – общее количество параметров; n – длина текущего вектора параметров; h_i – вектора слов; $a_{n,i}$ – объединяющая модель (нейросеть прямого распространения); $c_{n,i}$ – контекстный вектор; s_i – вектор декодируемых строк, сгенерированных из контекстных векторов слов; ULMfit – вектор признаков, полученный из универсальной языковой модели.

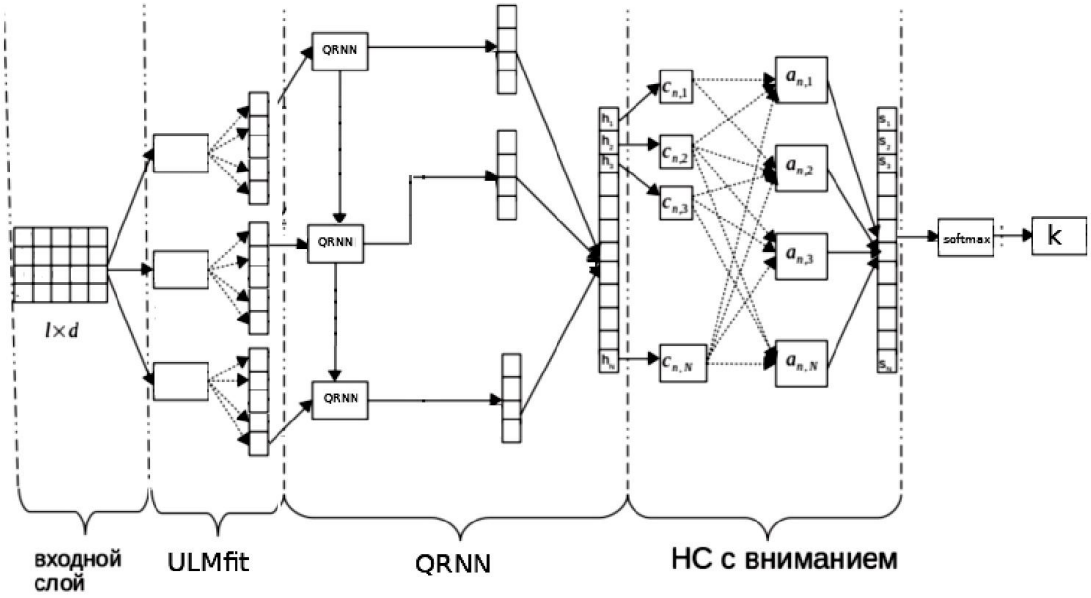

Рисунок 4 – Архитектура сети для классификации ошибок в тексте

При обучении модели классификации ошибок использовались следующие параметры: количество слоёв 5, функция активации скрытых слоев – ReLU, функция активации выходного слоя – Softmax, функция оптимизации – adam, функция потерь – кросс-энтропия (loss cse), размер скрытого слоя 512, количество эпох 10, размер пакета – 64, коэффициент dropout = 0.4.

Результаты численных исследований

В тестовый набор входили данные как без ошибок, так и содержащие ошибки разных классов.

Точность (accuracy) определялась следующим образом:

$$N_{right}/N,$$

где N_{right} – количество верно классифицированных ошибок; N – общее количество предложений.

В качестве оценок эффективности классификации ошибок обученной нейросетью предложенной архитектуры использовались значения функции потерь и accuracy (рис. 5, 6).

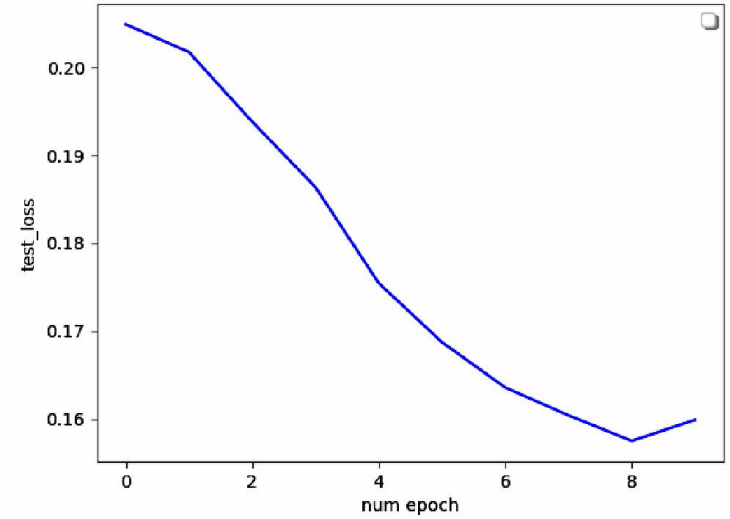

Рисунок 5 – Зависимость значений функции потерь на тестовой выборке от количества эпох обучения

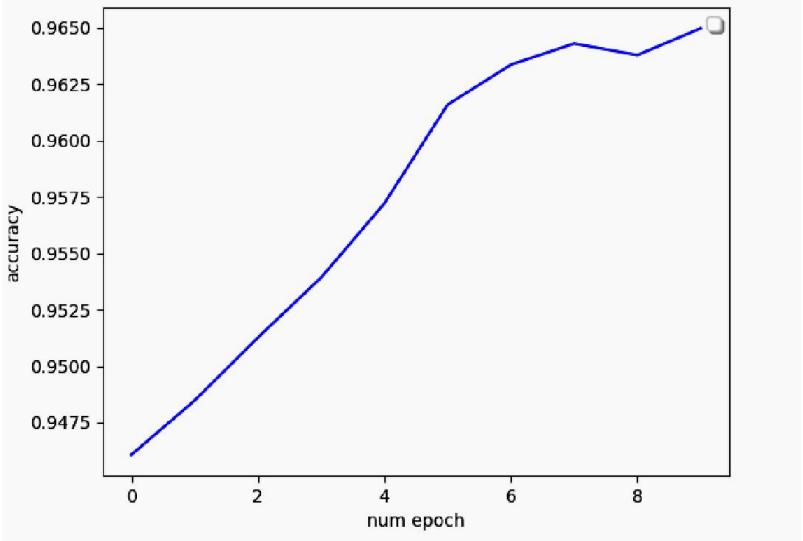

Рисунок 6 – Зависимость значений accuracy на тестовой выборке от количества эпох обучения

Для 100 случайно выбранных предложений из тестового набора на рис. 7 показаны гистограммы для верно определенных классов (True) и неверно определенных классов (False).

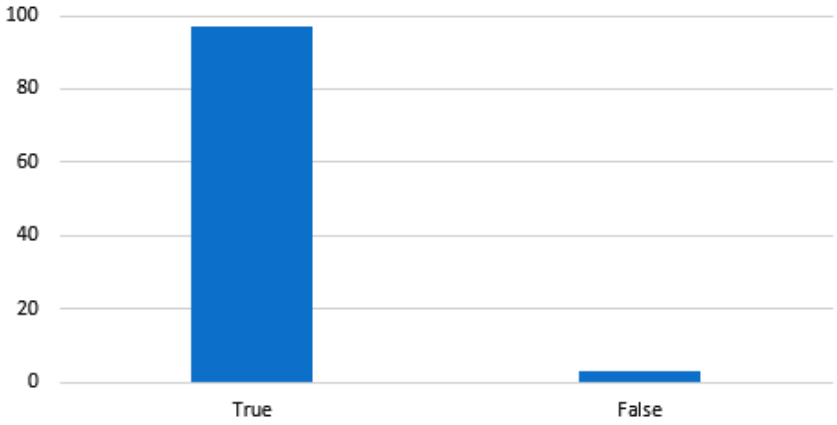

Рисунок 7 – Доли верно и неверно определённых классов

Выводы

Бурное развитие нейронных сетей, доступность больших корпусов текстовых данных для их обучения позволило использовать современные методы глубокого обучения для решения NLP-задач, к которым относится задача автоматической коррекции ошибок в текстах на естественном языке. Для задачи детектирования и классификации ошибок предложено векторное представление, основанное на архитектуре ULMfit, использующейся для построения универсальной языковой модели. Данный подход позволяет учитывать контекст слова, что немаловажно для детектирования ошибок в тексте.

100-мерный word embedding, полученный с выходов предпоследнего слоя ULMfit, поступает на сеть с архитектурой QRNN. Данная архитектура позволяет учитывать контекст и работает быстрее, чем стандартно используемые в NLP-задачах LSTM без потери точности. Как показали численные исследования, применение предложенного подхода к получению word embedding и использование архитектуры QRNN для классификации ошибок обеспечивают точность более 96%.

Список литературы

1. Шаврина Т. О. Методы обнаружения и исправления опечаток: исторический обзор [Текст] / Т. О. Шаврина // Вопросы языкознания. – 2017. – № 4. – С. 115-134.
2. «Диалог-2016», SpellRuEval [Электронный ресурс]. – URL: http://www.dialog-21.ru/media/3427/sorokinaaetal.pdf (дата обращения: 27.09.2019).
3. «Русский учебный корпус» [Электронный ресурс]. – URL: http://www.web-corpora.net/RLC (дата обращения: 27.09.2019).
4. Salton Gerard, Buckley Christopher. Term-weighting approaches in automatic text retrieval [Текст] // Information processing & management. – 1988. – Vol. 24, no. 5. – P. 513–523.
5. Tomas Mikolov et. al. Efficient Estimation of Word Representations in Vector Space [Электронный ресурс] // arXiv preprint arXiv:1301.3781. – 2013. URL: http://arxiv.org/pdf/1301.3781.pdf (дата обращения: 27.09.2019).

6. Joulin A. et al. Fasttext. zip: Compressing text classification models [Электронный ресурс] // arXiv preprint arXiv:1612.03651. – 2016. URL: https://arxiv.org/pdf/1612.03651.pdf (дата обращения: 27.09.2019).
7. Pennington J., Socher R., Manning C. Glove: Global vectors for word representation [Текст] // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). – 2014. – P. 1532–1543.
8. Jeremy Howard, Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification [Электронный ресурс] // arXiv preprint arXiv:1801.06146. – 2018. URL: http://arxiv.org/pdf/1801.06146.pdf (дата обращения: 27.09.2019).
9. Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models Classification [Электронный ресурс] // arXiv preprint arXiv: 1708.02182. – 2017. URL: arXiv preprint arXiv:1708.02182.pdf (дата обращения: 27.09.2019).
10. Николенко С. И. Глубокое обучение / С. И. Николенко, А. А. Кадурин, Е. В. Архангельская. – СПб.: Питер, 2018. – 480 с.
11. James Bradbury, Stephen Merity, Caiming Xiong, Richard Socher. Quasi-recurrent neural networks [Электронный ресурс] // arXiv preprint arXiv:1611.01576. – 2017. URL: https://arxiv.org/pdf/1611.01576.pdf (дата обращения: 27.09.2019).

References

1. Shavrina T.O. Metody` obnaruzheniya i ispravleniya opechatok: istoricheskij obzor [Methods for detecting and correcting typos: historical review]. *Voprosy` yazy`koznaniya* [Questions of linguistics], 2017, no 4, S. 115-134.
2. «Dialog-2016», SpellRuEval. URL: http://www.dialog-21.ru/media/3427/sorokinaaetal.pdf
3. «Russkij uchebny`j korpus», URL: http://www.web-corpora.net/RLC
4. Salton Gerard, Buckley Christopher. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 1988, Vol. 24, no. 5, P. 513–523.
5. Tomas Mikolov et. al. *Efficient Estimation of Word Representations in Vector Space*. arXiv preprint arXiv:1301.3781, 2013, URL: http://arxiv.org/pdf/1301.3781.pdf
6. Joulin A. et al. *Fasttext. zip: Compressing text classification models*. arXiv preprint arXiv:1612.03651, 2016, URL: https://arxiv.org/pdf/1612.03651.pdf
7. Pennington J., Socher R., Manning C. *Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, P. 1532–1543.
8. Jeremy Howard, Sebastian Ruder. *Universal Language Model Fine-tuning for Text Classification*. arXiv preprint arXiv:1801.06146, 2018, URL: http://arxiv.org/pdf/1801.06146.pdf (дата обращения: 27.09.2019).
9. Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. *Regularizing and optimizing LSTM language models Classification*. arXiv preprint arXiv: 1708.02182, 2017, URL: arXiv preprint arXiv:1708.02182.pdf (дата обращения: 27.09.2019).
10. Nikolenko S.I., Kadurin A.A., Arkhangel`skaya E.V. *Glubokoe obuchenie* [Deep learning], SPb., Piter, 2018, 480 s.
11. James Bradbury, Stephen Merity, Caiming Xiong, Richard Socher. *Quasi-recurrent neural networks*. arXiv preprint arXiv:1611.01576, 2017, URL: https://arxiv.org/pdf/1611.01576.pdf

RESUME

T. V. Yermolenko

Classification of Errors in the Text Based on Deep Learning

The rapid development of neural networks, the availability of large bodies of textual data for their training allowed the use of modern methods of deep learning to solve the NLP-tasks, which include the problem of automatic error correction in natural language texts. Automatic error correction, in addition to large vocabularies, using n-gram Language Model, built on large text corpora. Neural networks with LSTM architecture are used to account for the context.

The proposed approach for obtaining a vector representation of words uses a universal language model obtained on the basis of the ULMfit architecture. the text corpus on which the language model was trained, consisting of text arrays collected from news sites with a volume of 208 006 138 words, the total size of about 15 GB. To classify errors, a quasi-recurrent neural network using an attention layer is used.

As shown by numerical studies, applying the proposed approach to obtaining word embedding and using the QRNN architecture to classify errors provide an accuracy of more than 96%.

РЕЗЮМЕ

Т. В. Ермоленко

Классификация ошибок в тексте на основе глубокого обучения

На сегодняшний день для решения задач, связанных с обработкой естественно-языковых текстов используют методы глубокого обучения, что в первую очередь связано с ростом вычислительных мощностей и доступностью больших корпусов текстовых данных.

В данной статье предложена модель детектирования и классификации ошибок в тексте на основе глубокого обучения. Модель классификации ошибок основана на QRNN-архитектуре со слоем внимания, преимуществом которой является тот факт, что сеть работает в 16 раз быстрее, чем рекуррентные; в то же время показывая такую же точность классификации.

Для векторного представления слов используется предпоследний слой из обученной языковой модели на основе ULMfit. Данный подход позволяет учитывать контекст слова и лишен многих недостатков современных моделей векторного представления слов.

Для векторного представления слов использовался словарь, состоящий из 7 млн уникальных словоформ. Языковая модель обучалась на текстовый корпусе, состоящем из текстовых массивов, собранных из новостных сайтов объёмом в 208 006 138 слов, общим размером около 15 Гб.

Разработанные модели позволяют находить морфологические, орфографические и синтаксические ошибки с точностью более 96%.

Статья поступила в редакцию 25.07.2019.